

**LAPORAN PRAKTIKUM
PEMROGRAMAN MOBILE
MODUL 3**



BUILD A SCROLLABLE LIST

Oleh:

Muhammad Rizki Ramadhan

NIM. 2310817310008

**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS LAMBUNG MANGKURAT
MEI 2025**

LEMBAR PENGESAHAN
LAPORAN PRAKTIKUM PEMROGRAMAN I
MODUL 3

Laporan Praktikum Pemrograman Mobile Modul 2: Build a Scrollable List ini disusun sebagai syarat lulus mata kuliah Praktikum Pemrograman Mobile. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Muhammad Rizki Ramadhan
NIM : 2310817310008

Menyetujui,
Asisten Praktikum

Mengetahui,
Dosen Penanggung Jawab Praktikum

Muhammad Raka Azwar
NIM. 2210817210012

Andreyan Rizky Baskara, S.Kom., M.Kom.
NIP. 19930703 20190301011

DAFTAR ISI

LEMBAR PENGESAHAN	1
DAFTAR ISI	2
DAFTAR GAMBAR.....	3
DAFTAR TABEL	4
SOAL 1	6
A. Source Code.....	8
B. Output Program	26
C. Pembahasan	28
SOAL 2	35
D. Pembahasan	35
E. Tautan Git.....	35

DAFTAR GAMBAR

Gambar 1. UI List.....	6
Gambar 2. UI Detail	7
Gambar 4. UI List (XML)	26
Gambar 5. UI Detail (XML).....	26
Gambar 6. UI List (Jetpack Compose)	27
Gambar 7. UI Detail (Jetpack Compose).....	27

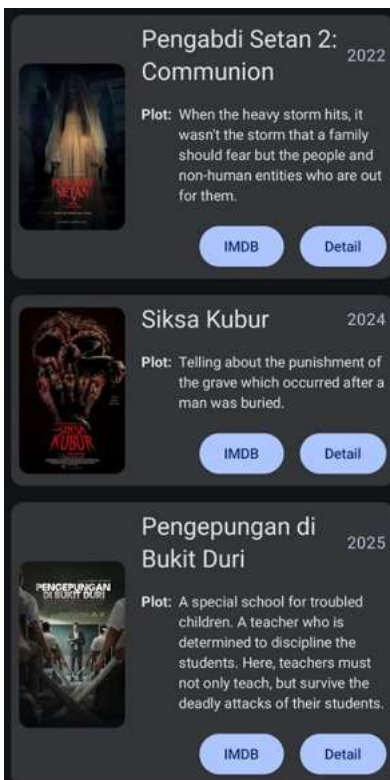
DAFTAR TABEL

Table 1. Source Code Jawaban Soal 1 XML.....	8
Table 2. Source Code Jawaban Soal 1 XML.....	8
Table 3. Source Code Jawaban Soal 1 XML.....	9
Table 4. Source Code Jawaban Soal 1 XM	11
Table 5. Source Code Jawaban Soal 1 XML.....	12
Table 6. Source Code Jawaban Soal 1 XML.....	13
Table 7. Source Code Jawaban Soal 1 XML.....	15
Table 8. Source Code Jawaban Soal 1 XML.....	17
Table 9. Source Code Jawaban Soal 1 Jetpack Compose.....	17
Table 10. Source Code Jawaban Soal 1 Jetpack Compose.....	19
Table 11. Source Code Jawaban Soal 1 Jetpack Compose.....	21
Table 12. Source Code Jawaban Soal 1 Jetpack Compose.....	23
Table 13. Source Code Jawaban Soal 1 Jetpack Compose.....	23
Table 14. Source Code Jawaban Soal 1 Jetpack Compose.....	24

SOAL 1

Buatlah sebuah aplikasi Android menggunakan XML dan Jetpack Compose yang dapat menampilkan list dengan ketentuan berikut:

1. List menggunakan fungsi RecyclerView (XML) dan LazyColumn (Compose)
2. List paling sedikit menampilkan 5 item. Tema item yang ingin ditampilkan bebas
3. Item pada list menampilkan teks dan gambar sesuai dengan contoh di bawah
4. Terdapat 2 button dalam list, dengan fungsi berikut:
 - a. Button pertama menggunakan intent eksplisit untuk membuka aplikasi atau browser lain
 - b. Button kedua menggunakan Navigation component untuk membuka laman detail item
5. Sudut item pada list dan gambar di dalam list melengkung atau rounded corner menggunakan Radius
6. Saat orientasi perangkat berubah/dirotasi, baik ke portrait maupun landscape, aplikasi responsif dan dapat menunjukkan list dengan baik. Data di dalam list tidak boleh hilang
7. Aplikasi menggunakan arsitektur single activity (satu activity memiliki beberapa fragment)
8. Aplikasi berbasis XML harus menggunakan ViewBinding



Gambar 1. UI List



Gambar 2. UI Detail

A. Source Code

XML:

MainActivity.kt

Table 1. Source Code Jawaban Soal 1 XML

```
package com.example.scrollablelistxml

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

class MainActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val recyclerView: RecyclerView =
            findViewById(R.id.recyclerView)
        recyclerView.layoutManager = LinearLayoutManager(this)
        recyclerView.adapter = ChocolateAdapter(this,
            DataProvider.chocolateList)
    }
}
```

Chocolate.kt

Table 2. Source Code Jawaban Soal 1 XML

```
package com.example.scrollablelistxml

data class Chocolate(
    val id: Int,
    val name: String,
    val description: String,
    val chocolateImageId: Int,
    val btnLink: String
)
```

ChocolateAdapter.kt

Table 3. Source Code Jawaban Soal 1 XML

```
package com.example.scrollablelistxml

import android.content.Context
import android.content.Intent
import android.net.Uri
import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.*
import androidx.recyclerview.widget.RecyclerView

class ChocolateAdapter(private val context: Context,
private val chocolateList: List<Chocolate>) :
    RecyclerView.Adapter<ChocolateAdapter.ViewHolder>() {

    class ViewHolder(view: View) :
        RecyclerView.ViewHolder(view) {
        val image: ImageView =
            view.findViewById(R.id.imageChocolate)
        val name: TextView =
            view.findViewById(R.id.textName)
        val description: TextView =
            view.findViewById(R.id.textDescription)
        val btnImdb: Button =
            view.findViewById(R.id.buttonImdb)
        val btnDetail: Button =
            view.findViewById(R.id.buttonDetail)
    }

    override fun onCreateViewHolder(parent: ViewGroup,
viewType: Int): ViewHolder {
        val view =
            LayoutInflater.from(context).inflate(R.layout.row_coklat,
parent, false)
        return ViewHolder(view)
    }

    override fun getItemCount(): Int = chocolateList.size
```

```

        override fun onBindViewHolder(holder: ViewHolder,
position: Int) {
            val chocolate = chocolateList[position]

holder.image.setImageResource(chocolate.chocolateImageId)
            holder.name.text = chocolate.name
            holder.description.text = chocolate.description

            holder.btnImdb.setOnClickListener {
                val intent = Intent(Intent.ACTION_VIEW,
Uri.parse(chocolate.btnLink))
                context.startActivity(intent)
            }

            holder.btnDetail.setOnClickListener {
                val intent = Intent(context,
DetailActivity::class.java).apply {
                    putExtra("id", chocolate.id)
                }
                context.startActivity(intent)
            }
        }
    }
}

```

DataAdapter.kt

Table 4. Source Code Jawaban Soal 1 XM

```
package com.example.scrollablelistxml

object DataProvider {
    val chocolateList = listOf(
        Chocolate(1, "Dark Chocolate", "Dark chocolate is
a form of chocolate made from cocoa solids, cocoa butter
and sugar. " +
            "It has a higher cocoa percentage than
white chocolate, milk chocolate, and semisweet chocolate.",
R.drawable.darkchocolate,
"https://en.wikipedia.org/wiki/Dark_chocolate"),
        Chocolate(2, "Milk Chocolate", "Milk chocolate is
a form of solid chocolate containing cocoa, sugar and milk.
" +
            "Milk chocolate contains smaller amounts of
cocoa solids than dark chocolates do, and (as with white
chocolate) contains milk solids.",
R.drawable.milkchocolate,
"https://en.wikipedia.org/wiki/Milk_chocolate"),
        Chocolate(3, "White Chocolate", "White chocolate is
a chocolate made from cocoa butter, sugar and milk solids.
" +
            "It is ivory in color and lacks the dark
appearance of most other types of chocolate as it does not
contain the non-fat components of cocoa (cocoa solids).",
R.drawable.whitechocolate,
"https://en.wikipedia.org/wiki/White_chocolate"),
        Chocolate(4, "Hazelnut Chocolate", "Hazelnut
chocolate adds a new dimension to the flavor by combining
the flavor of chocolate with the distinctive hazelnut nut.
" +
            "The hazelnut provides a savory, crunchy,
and slightly sweet touch that complements the richness of
the chocolate.", R.drawable.hazelnutchocolate,
"https://en.wikipedia.org/wiki/Chocolate#Flavors"),
        Chocolate(5, "Mint Chocolate", "The hazelnut
provides a savory, crunchy, and slightly sweet touch that
complements the richness of the chocolate. " +
```

```

        "Mint chocolate can be found in a wide
        variety of confectionery items, such as candy, mints,
        cookies, mint chocolate chip ice cream, hot chocolate, and
        others.
        ",
        R.drawable.mintchocolate,
        "https://en.wikipedia.org/wiki/Chocolate#Flavors")
    )
}

```

DetailActivity.kt

Table 5. Source Code Jawaban Soal 1 XML

```

package com.example.scrollablelistxml

import android.os.Bundle
import android.widget.ImageView
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity

class DetailActivity : AppCompatActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_detail)

        val id = intent.getIntExtra("id", -1)
        val chocolate = DataProvider.chocolateList.find {
            it.id == id }

        chocolate?.let {

            findViewById<ImageView>(R.id.imageChocolateDetail).setImage
            Resource(it.chocolateImageId)

            findViewById<TextView>(R.id.textTitleDetail).text = it.name

            findViewById<TextView>(R.id.textDescriptionDetail).text =
            it.description
        }
    }
}

```

activity_detail.kt

Table 6. Source Code Jawaban Soal 1 XML

```
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:padding="16dp">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content">

        <ImageView
            android:id="@+id/imageChocolateDetail"
            android:layout_width="0dp"
            android:layout_height="240dp"
            android:scaleType="centerCrop"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintEnd_toEndOf="parent" />

        <TextView
            android:id="@+id/textTitleDetail"
            android:layout_marginTop="16dp"
            android:textSize="24sp"
            android:textStyle="bold"
            android:layout_width="0dp"
            android:layout_height="wrap_content"

            app:layout_constraintTop_toBottomOf="@id/imageChocolateDetail"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintEnd_toEndOf="parent" />

        <TextView
            android:id="@+id/textLabelPlot"
            android:textStyle="bold"
            android:layout_marginTop="8dp"
            android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"

app:layout_constraintTop_toBottomOf="@id/textTitleDetail"
        app:layout_constraintStart_toStartOf="parent" />

    <TextView
        android:id="@+id/textDescriptionDetail"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp"

app:layout_constraintTop_toBottomOf="@id/textLabelPlot"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintBottom_toBottomOf="parent"
    />
    </androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>
```

Table 7. Source Code Jawaban Soal 1 XML

```
<androidx.cardview.widget.CardView
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:card_view="http://schemas.android.com/apk/res-auto"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="8dp"
    card_view:cardCornerRadius="12dp">

    <androidx.constraintlayout.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:padding="8dp">

        <com.google.android.material.imageview.ShapeableImageView
            android:id="@+id/imageChocolate"
            android:layout_width="100dp"
            android:layout_height="140dp"
            android:scaleType="centerCrop"
            android:layout_marginEnd="8dp"
            app:layout_constraintTop_toTopOf="parent"
            app:layout_constraintBottom_toBottomOf="parent"
            app:layout_constraintStart_toStartOf="parent"

            app:shapeAppearanceOverlay="@style/RoundedImageStyle" />

        <LinearLayout
            android:id="@+id/infoLayout"
            android:orientation="vertical"
            android:layout_width="0dp"
            android:layout_height="wrap_content"
            android:layout_marginStart="8dp"

            app:layout_constraintStart_toEndOf="@id/imageChocolate"
```



```

app:layout_constraintTop_toTopOf="@id/imageChocolate"

app:layout_constraintBottom_toBottomOf="@id/imageChocolate"
    app:layout_constraintEnd_toEndOf="parent">

    <TextView
        android:id="@+id/textName"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textStyle="bold"
        android:textSize="18sp" />

    <TextView
        android:id="@+id/textDescription"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="4dp" />

    <LinearLayout
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:orientation="horizontal"
        android:layout_marginTop="8dp">

        <Button
            android:id="@+id/buttonImdb"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="IMDB" />

        <Button
            android:id="@+id/buttonDetail"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Detail"
            android:layout_marginStart="8dp" />
    </LinearLayout>
</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

```

```
</androidx.cardview.widget.CardView>
```

activity_main.xml

Table 8. Source Code Jawaban Soal 1 XML

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <androidx.recyclerview.widget.RecyclerView
        android:id="@+id/recyclerView"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>
</LinearLayout>
```

Jetpack Compose:

MainActivity.kt

Table 9. Source Code Jawaban Soal 1 Jetpack Compose

```
package com.example.scrollablelistcompose

import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.layout.padding
import androidx.compose.material3.Scaffold
import androidx.compose.runtime.Composable
import androidx.compose.runtime.remember
import androidx.compose.ui.Modifier
import androidx.navigation.NavType
import androidx.navigation.compose.NavHost
import androidx.navigation.compose.composable
import androidx.navigation.compose.rememberNavController
import androidx.navigation.navArgument
import com.example.scrollablelistcompose.data.DataProvider
```

```

import
com.example.scrollablelistcompose.ui.theme.ScrollAbleListComposeTheme

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            ScrollAbleListComposeTheme {
                val navController = rememberNavController()
                MyApp(navController = navController)
            }
        }
    }
}

@Composable
fun MyApp(navController: androidx.navigation.NavHostController) {
    val chocolates = remember { DataProvider.chocolateList }

    Scaffold { innerPadding ->
        NavHost(
            navController = navController,
            startDestination = "home",
            modifier = Modifier.padding(innerPadding)
        ) {
            composable("home") {
                ScrollAbleListHomeContent(
                    onDetailClick = { selected ->
                        navController.navigate("detail/${selected.no}")
                    }
                )
            }
            composable(
                "detail/{id}",
                arguments = listOf(navArgument("id") { type =
NavType.IntType })
            ) { backStackEntry ->
                val id = backStackEntry.arguments?.getInt("id")
                val chocolate = chocolates.find { it.no == id }
                chocolate?.let {
                    DetailScreen(it)
                }
            }
        }
    }
}

```

```
}  
}
```

ChocolateListItem.kt

Table 10. Source Code Jawaban Soal 1 Jetpack Compose

```
package com.example.scrollablelistcompose  
  
import android.content.Intent  
import android.net.Uri  
import androidx.compose.foundation.Image  
import androidx.compose.foundation.layout.Column  
import androidx.compose.foundation.layout.Row  
import androidx.compose.foundation.layout.Spacer  
import androidx.compose.foundation.layout.fillMaxWidth  
import androidx.compose.foundation.layout.height  
import androidx.compose.foundation.layout.padding  
import androidx.compose.foundation.layout.width  
import androidx.compose.foundation.shape.RoundedCornerShape  
import androidx.compose.material3.Button  
import androidx.compose.material3.Card  
import androidx.compose.material3.MaterialTheme  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.draw.clip  
import androidx.compose.ui.layout.ContentScale  
import androidx.compose.ui.platform.LocalContext  
import androidx.compose.ui.res.painterResource  
import androidx.compose.ui.unit.dp  
import com.example.scrollablelistcompose.data.Chocolate  
  
@Composable  
fun ChocolateListItem(chocolate: Chocolate, onDetailClick:  
    (Chocolate) -> Unit) {  
    val context = LocalContext.current  
  
    Card(  
        modifier = Modifier  
            .padding(8.dp)  
            .fillMaxWidth(),
```

```

        shape = RoundedCornerShape(16.dp)
    ) {
        Row(
            modifier = Modifier.padding(8.dp),
            verticalAlignment = Alignment.CenterVertically
        ) {

            Image(
                painter = painterResource(id =
chocolate.chocolateImageId),
                contentDescription = null,
                contentScale = ContentScale.Crop,
                modifier = Modifier
                    .width(100.dp)
                    .height(140.dp)
                    .clip(RoundedCornerShape(12.dp))
            )

            Spacer(modifier = Modifier.width(12.dp))

            Column(
                modifier = Modifier
                    .padding(16.dp)
                    .weight(1f)
            ) {
                Text(text = "${chocolate.title}")
                Text(text = "No: ${chocolate.no}")

                Text(
                    text = "${chocolate.description}",
                    style = MaterialTheme.typography.bodySmall,
                    modifier = Modifier.padding(top = 4.dp)
                )

                Row(modifier = Modifier.padding(top = 8.dp)) {
                    Button(onClick = {
                        val intent = Intent(Intent.ACTION_VIEW,
Uri.parse(chocolate.btnLink))
                        context.startActivity(intent)
                    }) {
                        Text("IMDB")
                    }
                }
            }
        }
    }
}

```

```

        Spacer(modifier = Modifier.padding(horizontal =
4.dp))

        Button(onClick = {
            onDetailClick(chocolate)

        }) {
            Text("Detail")
        }
    }
}

}

}

}

@Composable
fun ChocolateImage(chocolate: Chocolate, modifier: Modifier =
Modifier) {
    Image(
        painter = painterResource(id = chocolate.chocolateImageId),
        contentDescription = null,
        contentScale = ContentScale.Crop,
        modifier = Modifier
            .padding(8.dp)
            .width(120.dp)
            .height(150.dp)
            .clip(RoundedCornerShape(16.dp))
    )
}

```

DetailScreen.kt

Table 11. Source Code Jawaban Soal 1 Jetpack Compose

```
package com.example.scrollablelistcompose

import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.Column
import androidx.compose.foundation.layout.Row
import androidx.compose.foundation.layout.Spacer
import androidx.compose.foundation.layout.fillMaxWidth
import androidx.compose.foundation.layout.height
import androidx.compose.foundation.layout.padding
import androidx.compose.foundation.shape.RoundedCornerShape
import androidx.compose.material3.MaterialTheme
```

```

import androidx.compose.material3.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.clip
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.unit.dp
import com.example.scrollablelistcompose.data.Chocolate

@Composable
fun DetailScreen(chocolate: Chocolate) {
    Column(modifier = Modifier
        .padding(16.dp)
        .fillMaxWidth()
    ) {
        Image(
            painterResource(id = R.drawable.chocolate_image),
            chocolate.chocolateImageId,
            contentDescription = null,
            contentScale = ContentScale.Crop,
            modifier = Modifier
                .fillMaxWidth()
                .height(240.dp)
                .clip(RoundedCornerShape(12.dp))
        )

        Spacer(modifier = Modifier.height(16.dp))

        Row(
            modifier = Modifier.fillMaxWidth(),
            verticalAlignment = Alignment.CenterVertically
        ) {
            Text(
                text = chocolate.title,
                style = MaterialTheme.typography.titleLarge,
                modifier = Modifier.weight(1f)
            )
        }

        Spacer(modifier = Modifier.height(8.dp))

        Text(
            text = chocolate.description,
            style = MaterialTheme.typography.bodyMedium
        )
    }
}

```

```

    )
}
}

```

ScrollAbleListHome.kt

Table 12. Source Code Jawaban Soal 1 Jetpack Compose

```

package com.example.scrollablelistcompose

import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.items
import androidx.compose.runtime.remember
import androidx.compose.runtime.Composable
import com.example.scrollablelistcompose.data.DataProvider
import androidx.compose.ui.Modifier
import com.example.scrollablelistcompose.data.Chocolate

@Composable
fun ScrollAbleListHomeContent(modifier: Modifier = Modifier,
onDetailClick: (Chocolate) -> Unit = {}) {
    val chocolates = remember { DataProvider.chocolateList }
    LazyColumn(modifier = modifier) {
        items(chocolates) { chocolate ->
            ChocolateListItem(chocolate, onDetailClick =
{onDetailClick(it)})
        }
    }
}

```

Chocolate.kt

Table 13. Source Code Jawaban Soal 1 Jetpack Compose

```

package com.example.scrollablelistcompose.data

data class Chocolate(
    val no: Int,
    val title: String,
    val description: String,
    val chocolateImageId: Int,
    val btnLink: String
)

```


DataProvider.kt

Table 14. Source Code Jawaban Soal 1 Jetpack Compose

```
package com.example.scrollablelistcompose.data

import com.example.scrollablelistcompose.R

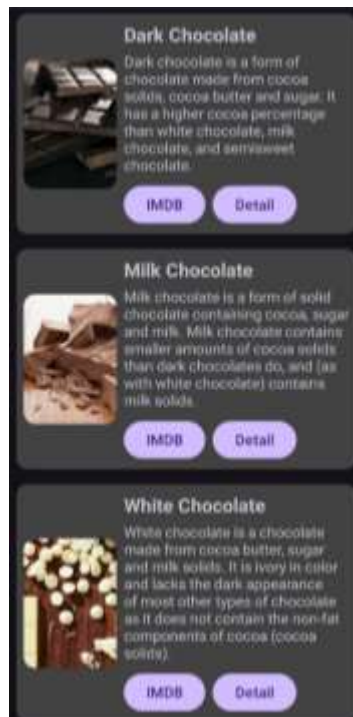
object DataProvider {
    val chocolateList = listOf(
        Chocolate(
            no = 1,
            title = "Dark Chocolate",
            description = "Dark chocolate is a form of chocolate made from cocoa solids, cocoa butter and sugar. " +
                "It has a higher cocoa percentage than white chocolate, milk chocolate, and semisweet chocolate.",
            chocolateImageId = R.drawable.darkchocolate,
            btnLink = "https://en.wikipedia.org/wiki/Dark_chocolate"
        ),
        Chocolate(
            no = 2,
            title = "Milk Chocolate",
            description = "Milk chocolate is a form of solid chocolate containing cocoa, sugar and milk. " +
                "Milk chocolate contains smaller amounts of cocoa solids than dark chocolates do, and (as with white chocolate) contains milk solids.",
            chocolateImageId = R.drawable.milkchocolate,
            btnLink = "https://en.wikipedia.org/wiki/Milk_chocolate"
        ),
        Chocolate(
            no = 3,
            title = "White Chocolate",
            description = "White chocolate is a chocolate made from cocoa butter, sugar and milk solids. " +
                "It is ivory in color and lacks the dark appearance of most other types of chocolate as it does not contain the non-fat components of cocoa (cocoa solids).",
            chocolateImageId = R.drawable.whitechocolate,
            btnLink = "https://en.wikipedia.org/wiki/White_chocolate"
        ),
        Chocolate(
```

```

        no = 4,
        title = "Hazelnut Chocolate",
        description = "Hazelnut chocolate adds a new dimension
to the flavor by combining the flavor of chocolate with the
distinctive hazelnut nut. " +
            "The hazelnut provides a savory, crunchy, and
slightly sweet touch that complements the richness of the
chocolate.",
        chocolateImageId = R.drawable.huzelnutchocolate,
        btnLink =
"https://kioskcokelat.com/blogs/news/perbedaan-coklat-dan-coklat-
hazelnut"
    ),
    Chocolate(
        no = 5,
        title = "Mint Chocolate",
        description = "Mint chocolate (or chocolate mint) is a
popular flavor of chocolate, made by adding a mint flavoring, such
as peppermint, spearmint, or crème de menthe, to chocolate. " +
            "Mint chocolate can be found in a wide variety
of confectionery items, such as candy, mints, cookies, mint chocolate
chip ice cream, hot chocolate, and others.",
        chocolateImageId = R.drawable.mintchocolate,
        btnLink =
"https://en.wikipedia.org/wiki/Chocolate#Flavors"
    )
)
}

```

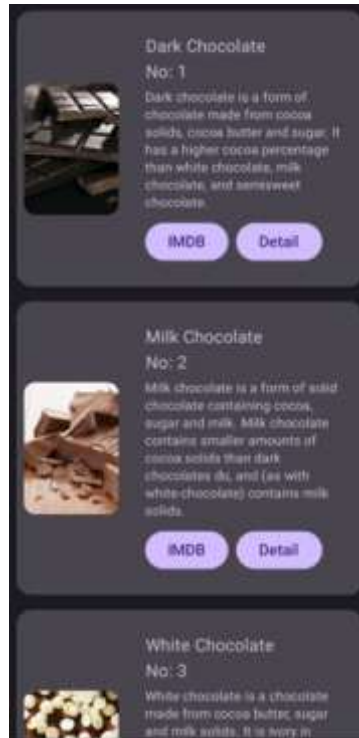
B. Output Program



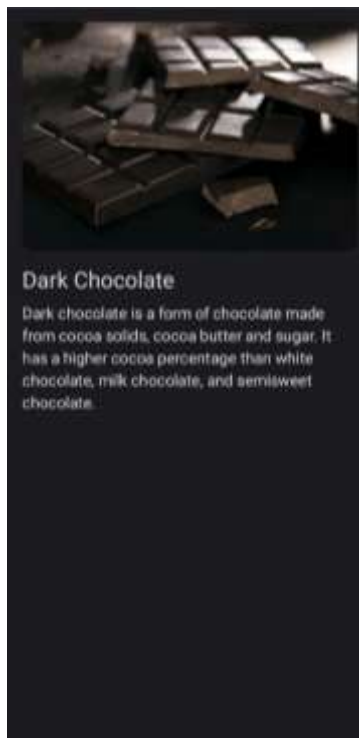
Gambar 3. UI List (XML)



Gambar 4. UI Detail (XML)



Gambar 5. UI List (Jetpack Compose)



Gambar 6. UI Detail (Jetpack Compose)

C. Pembahasan

MainActivity.kt (XML):

1. Syntax `package package com.example.scrollablelistxml` merupakan deklarasi nama package file Kotlin.
2. Syntax `class MainActivity : AppCompatActivity() {}` merupakan deklarasi kelas bernama MainActivity yang extend dari AppCompatActivity.
3. Syntax `private lateinit var binding: ActivityMainBinding` mendeklarasikan objek binding untuk mengakses view dari XML via viewBinding.
4. Syntax `override fun onCreate(savedInstanceState: Bundle?) {}` merupakan lifecycle method pertama yang dipanggil saat Activity dibuat.
5. Syntax `super.onCreate(savedInstanceState)` merupakan program untuk memanggil method onCreate() dari superclass untuk menjalankan proses inisialisasi default dan wajib dipanggil agar activity dapat berjalan dengan benar.
6. Syntax `setContentView(R.layout.activity_main)` program untuk menetapkan layout XML yang digunakan sebagai UI untuk activity ini.
7. Syntax `val recyclerView: RecyclerView = findViewById(R.id.recyclerView)` merupakan fungsi untuk mendeklarasikan dan menginisialisasi variabel recyclerView dari layout XML, findViewById mencari view berdasarkan IDE @+id/recyclerView yang telah didefinisikan di XML.
8. Syntax `recyclerView.layoutManager = LinearLayoutManager(this)` berfungsi untuk mengatur adapter untuk RecyclerView, adapter ChocolateAdapter bertanggung jawab untuk menghubungkan data (chocolateList) dengan tampilan list, this mengacu pada konteks activity, dan DataProvider.chocolateList adalah sumber data.

Chocolate.kt (XML):

1. Syntax `package package com.example.scrollablelistxml` merupakan deklarasi package tempat file ini berada, yaitu com.example.scrollablelistxml.
2. Syntax `data class Chocolate()` merupakan definisi data class bernama Chocolate, data class secara otomatis menyediakan fungsi-fungsi penting, class ini biasanya dipakai untuk menyimpan data sederhana, seperti model atau entitas.
3. Syntax `val id: Int, val name: String, val description: String, val chocolateImageId: Int, val btnLink: String` mendeklarasikan variabel dengan tipe datanya.

ChocolateAdapter.kt (XML):

1. Syntax `class ChocolateAdapter(...)` :
`RecyclerView.Adapter<ChocolateAdapter.ViewHolder>()`
membuat class `ChocolateAdapter`, turunan dari `RecyclerView.Adapter`. Adapter ini digunakan untuk menjembatani data `Chocolate` dengan tampilan di `RecyclerView`.
2. Syntax `class ViewHolder (view: View)` :
`RecyclerView.ViewHolder (view)` merupakan wadah (wrapper) untuk setiap item yang akan ditampilkan di `RecyclerView`.
3. Syntax `val image: ImageView` =
`view.findViewById(R.id.imageChocolate)` menghubungkan view dari XML (`R.id.imageChocolate`) ke variabel `image`.
4. Syntax `val name: TextView` =
`view.findViewById(R.id.textName)` merupakan program untuk menghubungkan `TextView` dengan cokelat.
5. Syntax `val description: TextView` =
`view.findViewById(R.id.textDescription)` merupakan program untuk untuk menyambungkan `TextView` deskripsi cokelat.
6. Syntax `val btnImdb: Button` =
`view.findViewById(R.id.buttonImdb)` program untuk menghubungkan tombol `IMDb` atau link ke variabel `btnImdb`.
7. Syntax `val btnDetail: Button` =
`view.findViewById(R.id.buttonDetail)` merupakan fungsi untuk menghubungkan tombol `Detail` ke variabel `btnDetail`.
8. Syntax `override fun onCreateViewHolder(...)` berfungsi untuk mengubah `row_coklat.xml` menjadi objek view dan membuat `ViewHolder` dari tampilan tersebut.
9. Syntax `val chocolate = chocolateList[position]` merupakan program untuk mengambil objek `Chocolate` berdasarkan indeks.
10. Syntax
`holder.image.setImageResource(chocolate.chocolateImageId)`
) menetapkan gambar dari resource ID ke `ImageView`.
11. Syntax `holder.name.text = chocolate.name` menampilkan nama cokelat.
12. Syntax `holder.btnImdb.setOnClickListener { ... }` merupakan program untuk menetapkan aksi saat tombol `IMDb` diklik dan membuat intent dengan `ACTION_VIEW` untuk membuka `chocolate.btnLink` di browser.
13. Syntax `holder.btnDetail.setOnClickListener { ... }` merupakan aksi untuk tombol `Detail`, membuat `Intent` untuk membuka `DetailActivity`, menyisipkan id dari cokelat sebagai data ekstra (`putExtra("id", chocolate.id)`).

DataProvider.kt (XML):

1. Syntax `object DataProvider {` merupakan program singleton object bernama `DataProvider`.
2. Syntax `val chocolateList = listOf(...)` mendeklarasikan properti `chocolateList` bertipe `List` dari objek `Chocolate`, menggunakan `val` karena list-nya bersifat immutable (tidak dapat diganti dengan list baru), dan fungsi `listOf(...)` digunakan untuk membuat list berisi data-data `Chocolate`.

DetailActivity.kt (XML):

1. Syntax `package package com.example.scrollablelistxml` merupakan deklarasi nama package file Kotlin.
2. Syntax `class DetailActivity : AppCompatActivity() {}` merupakan deklarasi kelas bernama `DetailActivity` yang extend dari `AppCompatActivity`.
3. Syntax `override fun onCreate(savedInstanceState: Bundle?) {}` mendeklarasikan fungsi `onCreate` dipanggil saat activity dibuat pertama kali, parameter menyimpan state sebelumnya jika ada.
4. Syntax `super.onCreate(savedInstanceState)` merupakan fungsi untuk memanggil implementasi `onCreate` dari superclass (`AppCompatActivity`), wajib dilakukan agar lifecycle Android tetap berjalan normal.
5. Syntax `setContentView(R.layout.activity_detail)` menetapkan layout XML `activity_detail.xml` sebagai tampilan utama activity ini.
6. Syntax `val id = intent.getIntExtra("id", -1)` mengambil data id yang dikirim dari `ChocolateAdapter` via `Intent`, jika id tidak ditemukan, default-nya adalah `-1`.
7. Syntax `val chocolate = DataProvider.chocolateList.find { it.id == id }` merupakan Mencari objek `Chocolate` dalam `DataProvider.chocolateList` yang ID-nya cocok dengan data id dari intent, menggunakan fungsi `find { ... }`, yang akan mengembalikan elemen pertama yang cocok atau null jika tidak ditemukan.
8. Syntax `chocolate?.let { ... }` berfungsi untuk mengecek apakah `chocolate` tidak null, jika ya, maka blok `let` akan dijalankan untuk menampilkan data.
9. Syntax `findViewById<ImageView>(R.id.imageChocolateDetail).setImageResource(it.chocolateImageId)` berfungsi untuk mengatur gambar pada `ImageView` di layout detail berdasarkan gambar dari objek `Chocolate`.

10. Syntax `findViewById<TextView>(R.id.textTitleDetail).text = it.name` berfungsi untuk menampilkan nama coklat ke TextView dengan ID `textTitleDetail`.
11. Syntax `findViewById<TextView>(R.id.textDescriptionDetail).text = it.description` berfungsi untuk menampilkan deskripsi coklat ke TextView dengan ID `textDescriptionDetail`.

activity_detail.xml

1. Syntax `package ScrollView` merupakan komponen yang memungkinkan isi layout dapat digulir secara vertikal jika kontennya lebih tinggi dari layar.
2. Syntax `ConstraintLayout` merupakan layout fleksibel dan powerful yang memungkinkan kita mengatur posisi elemen UI berdasarkan hubungan (constraint) antar elemen atau dengan parent.
3. Syntax `ImageView` untuk menampilkan gambar.
4. Syntax `TextView` merupakan program untuk menampilkan nama coklat, deskripsi.

activity_main.xml

1. Syntax `<LinearLayout>` merupakan layout linear yang menyusun elemen-elemen UI secara vertikal (satu di atas yang lain) karena `orientation="vertical"`.
2. Syntax `RecyclerView` merupakan komponen yang digunakan untuk menampilkan daftar item (seperti daftar coklat) secara efisien dan bisa discroll. Syntax `ImageView` untuk menampilkan gambar.

MainActivity.kt (Jetpack Compose):

1. Syntax `package package com.example.scrollablelistcompose` merupakan deklarasi nama package file Kotlin.
2. Syntax `class MainActivity : ComponenttActivity() {}` merupakan activity untuk Jetpack Compose yang berisi program UI penuh di layer, mengatur UI dengan compose, dan program untuk tema dan memanggil composable utama.
3. Syntax `override fun onCreate(savedInstanceState: Bundle?) {super.onCreate(savedInstanceState)}` merupakan override fungsi `onCreate` untuk menginisialisasi UI saat activity dibuat.
4. Syntax `ScrollAbleListComposeTheme{}` merupakan program untuk `(screen)`.
5. Syntax `MyApp(navController = navController)` untuk menjalankan composable `MyApp()` dan meneruskan `navController`.
6. Syntax `@Composable fun MyApp(navController: androidx.navigation.NavHostController) {` merupakan fungsi composable utama yang menerima `NavController` untuk navigasi antar layar.
7. Syntax `Scaffold { innerPadding -> }` merupakan program untuk menyediakan struktur dasar seperti padding otomatis, top bar, FAB, dsb. Kemudian `innerPadding` adalah padding otomatis agar konten tidak tertutup oleh elemen UI sistem seperti status bar atau navigation bar.
8. Syntax `NavHost() {}` merupakan program untuk mengelola semua screen dan navigation berdasarkan `navController`. Dan `startDestination` adalah layar pertama yang akan muncul, yaitu "home".
9. Syntax `composable("home") {}` merupakan program untuk jika item di-klik, akan berpindah ke screen "detail/{id}" dengan mengirimkan no dari objek yang diklik.
10. Syntax `composable()` merupakan program untuk mendefinisikan rute dengan argumen: "detail/{id}", di mana id adalah Int.

ChocolateListItem.kt (Jetpack Compose):

1. Syntax `package package com.example.scrollablelistcompose` merupakan deklarasi nama package file Kotlin.
2. Syntax `@Composable fun ChocolateListItem(chocolate: Chocolate, onDetailClick: (Chocolate) -> Unit) {` merupakan program untuk menandai fungsi sebagai composable.
3. Syntax `Card () {` merupakan program untuk latar belakang.

4. Syntax `Row() {` merupakan row utama untuk mengatur konsen dengan berbagai ketentuan.
5. Syntax `Image()` merupakan fungsi untuk menampilkan gambar dengan resource ID.
6. Syntax `Column() {}` merupakan program untuk mengatur teks dan tombol pada aplikasi.
7. Syntax `@Composable fun ChocolateImage(chocolate: Chocolate, modifier: Modifier = Modifier) {` merupakan program untuk menampilkan gambar.

ScrollableListHome.kt (Jetpack Compose):

1. Syntax `package package com.example.scrollablelistcompose` merupakan deklarasi nama package file Kotlin.
2. Syntax `@Composable fun ScrollableListHomeContent(modifier: Modifier = Modifier, onDetailClick: (Chocolate) -> Unit = {})` {merupakan program untuk menandai fungsi sebagai composable dan menerima dua parameter untuk styling dan klik tombol.
3. Syntax `val chocolates = remember {`
`DataProvider.chocolateList }` merupakan program untuk mengambil daftar coklat dari `DataProvider` dan agar data tidak diambil ulang setiap kali recomposition terjadi.
4. Syntax `LazyColumn(modifier = modifier) {` merupakan list scrollable vertikal yang hanya memuat item yang terlihat di layar (efisien).

Chocolate.kt (Jetpack Compose):

1. Syntax `package package com.example.scrollablelistcompose` merupakan deklarasi nama package file Kotlin.
2. Syntax `data class Chocolate(`
`val no: Int,`
`val title: String,`
`val description: String,`
`val chocolateImageId: Int,`
`val btnLink: String`
merupakan kelas untuk mendefinisikan kelas yang terutama digunakan untuk menyimpan data. Kotlin secara otomatis menyediakan beberapa fungsi tambahan seperti `toString()`, `equals()`, `hashCode()`, dan `copy()` untuk kelas ini.

DataProvider.kt (Jetpack Compose):

1. Syntax `package package com.example.scrollablelistcompose` merupakan deklarasi nama package file Kotlin.
2. Syntax `object DataProvider {}` merupakan object Kotlin, yang berfungsi sebagai singleton (hanya ada satu instance). Objek ini digunakan untuk menyediakan data statis, dalam hal ini adalah daftar coklat.
3. Syntax `val chocolateList = listOf()` merupakan properti yang berisi daftar objek Chocolate. Menggunakan `listOf()` untuk membuat list dari beberapa objek Chocolate.
4. Syntax `Chocolate()`, merupakan objek Chocolate yang mendefinisikan sebuah coklat lengkap dengan ID, nama, deskripsi, gambar, dan link.

SOAL 2

Mengapa RecyclerView masih digunakan, padahal RecyclerView memiliki kode yang panjang dan bersifat boiler-plate, dibandingkan LazyColumn dengan kode yang lebih singkat?

D. Pembahasan

Karena RecyclerView masih kompatibilitas saat ini, terlebih keterbatasan LazyColumn untuk kasus-kasus spesifik, seperti pagination yang kompleks, data binding tingkat lanjut, multi-view types, dan item click listeners. Kemudian RecyclerView juga dapat memberikan lebih banyak kontrol terkait optimasi memori dengan fitur seperti ViewPool, yang memungkinkan daftar item tampilan yang sudah ada untuk meningkatkan kinerja. Selain itu, perpindahan dari XML ke Jetpack Compose juga memerlukan waktu dan perubahan besar dalam struktur aplikasi apalagi aplikasi yang dari awal dikembangkan dengan XML yang membutuhkan pemahaman dan penguasaan baru untuk bisa berpindah menggunakan Jetpack Compose.

E. Tautan Git

Berikut adalah tautan untuk source code yang telah dibuat.

<https://github.com/rizkiirr/PemrogramanMobile.git>